

# Interaction with digital objects in JavaScript

Pere Colomer Gallardo

**Resum—** Interaction with digital objects in JavaScript és un projecte de final de grau que es basa en la implementació d'un software interactiu multimèdia. Aquest software consta d'una interfície gràfica intuïtiva on l'usuari pot crear animacions digitals en JavaScript a partir de la llibreria Paper.JS. Aquest software permet crear i interactuar amb unes figures definides en la llibreria, pot crear diferents escenes de l'animació i modificar les figures en les diferents escenes. Aquestes modificacions poden ser de mida, posició, rotació o color. L'usuari podrà introduir el temps que desitja que transcorri entre escena i escena. El programari també tindrà la capacitat d'afegir so a l'animació. D'aquesta manera, Interaction with digital objects in JavaScript ens permet crear animacions digitals on el codi resultant d'aquesta animació serà JavaScript.

**Paraules clau—** Animació digital, JavaScript, Paper.js, html5, css, escenes, canvas.

**Abstract—** Interaction with digital objects in Javascript is a degree's final project based on the implementation of an interactive multimedia software. This software consists of an intuitive UI where the user can create digital animations in JS using the Paper.JS library. This software allows the creation and interaction with the figures defined in the library and it can create different scenes from the animation or modify said figures. These modifications may be size wise, position wise or color wise. The user may input the time to pass between scenes. The software will also include a choice to add sound to the animation. Thus, Interaction with digital objects in JS aids in the creation of animations where the resulting code will be entirely JS.

**Keywords—** Digital animation, JavaScript, Paper.js, html5, css, scenes, canvas.

## 1 INTRODUCTION

L A idea del projecte sorgeix de la necessitat per part del Centre de Visió per Computació, de, en el seu projecte de eye tracking, tenir una eina que sigui capaç de crear animacions amb objectes digitals mitjançant el llenguatge JavaScript per, posteriorment, utilitzar el codi JavaScript generat per poder interaccionar amb aquests objectes. Per satisfer aquesta necessitat es decideix implementar un software que ens permeti crear animacions on el codi resultant sigui JavaScript i on aquest codi pugui ser utilitzat en altres aplicacions. Es marquen diferents objectius en la captura de requisits d'aquest projecte, primer de tot, existeix la necessitat de disposar d'una llibreria JavaScript que ens permeti realitzar el software, s'avaluen les diferents possibilitats ja existents en el mercat i es decideix utilitzar la llibreria Paper.JS. Els objectius del projecte doncs, eren crear una aplicació web on es pogués crear una animació amb diferents elements digitals on

aquests anaven sent modificats al llarg del temps, aquestes modificacions havien de poder ser de mida, posició, color o rotació. Tot això acompanyat d'un so que l'usuari podia introduir a l'aplicació.

En aquest document s'explicaran els passos realitzats per complir amb els objectius inicials. En primer lloc es realitzarà una petita explicació de l'estat de l'art, és a dir, què hi havia al mercat abans de començar el desenvolupament, a continuació s'explicarà la metodologia utilitzada per dur a terme el projecte (metodologia, eines i planificació). S'exposarà el disseny que es va crear per l'aplicació web tot seguit dels resultats obtinguts. Un cop haguem vist els resultats obtinguts s'exposaran les possibles millores d'aquesta aplicació així com idees a seguir per augmentar les funcionalitats d'aquesta. Per acabar, s'exposaran les conclusions extretes durant el llarg del treball.

## 2 ESTAT DE L'ART

Actualment existeixen diverses eines que poden ajudar en la implementació del treball:

- E-mail de contacte: pere.colomer@e-campus.uab.cat
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Fernando Vilariño (CVC)
- Curs 2015/16

## 2.1 Canvas de HTML5

Canvas és un element introduït en HTML5 que permet la generació de gràfics dinàmicament per mitjà de scripts en JavaScript. Ens permet dibuixar gràfics estàtics i animacions en una pàgina web [7].

## 2.2 Paper.js

Paper.js és una llibreria open source de JavaScript desenvolupada per Jürg Lehni i Jonathan Puckey. Ens permet la construcció de gràfics utilitzant l'element html canvas. Ofereix una elevada diversitat d'escenes gràfiques i models d'objectes amb un gran número de funcionalitats per poder crear i treballar amb gràfics. Aquesta llibreria proporciona facilitat a l'usuari a l'hora de treballar amb gràfics en JavaScript, per tant, també ens permetria adaptar-ho als nostres objectius per tal de poder crear animacions a partir dels objectes i escenaris que ens presenta[1][4].

## 2.3 Flocking.js

Flocking.js és una llibreria de JavaScript utilitzada per la síntesi d'àudio, està dissenyada per a artistes i músics que estan construint projectes de so basats en web. Flocking ens permet utilitzar diferents sons ja definits en la nostre web i escollir diferents paràmetres com poden ser el volum o la freqüència[16].

Com poder comprobar doncs, al mercat ja existeixen eines que permeten crear animacions web en JavaScript, el problema però, és que no existeix un software que ens permeti utilitzar aquestes eines sense tenir coneixements de programació JavaScript, i per tant, un usuari que no coneixi JavaScript i no coneixi aquestes eines, no pot dur a terme animacions. Així doncs, el nostre projecte té l'objectiu d'ocupar aquest lloc en el mercat, és a dir, que un usuari sense coneixements previs en programació sigui capaç de crear una animació web, on el resultat resultant sigui JavaScript i on aquest codi pugui ser utilitzat posteriorment en altres aplicacions.

## 3 METODOLOGIA

A continuació s'explicarà la metodologia aplicada durant el projecte i també quines eines s'han utilitzat per dur a terme aquesta metodologia.

### 3.1 Elicitació i requisits de l'aplicació

En aquest projecte han existit dos stakeholders principals, primer de tot el tutor del projecte Fernando Vilarino, del centre de visió per computació, i per altre banda l'artista Dan Norton [5], col·laborador d'aquest projecte. D'aquesta manera, amb les reunions realitzades al principi del projecte amb els stakeholders es va aconseguir extreure el llistat de requeriments necessaris a desenvolupar durant el projecte. A partir d'aquests requeriments es va realitzar un llistat d'objectius del projecte i la seva corresponent planificació. A més a més, de les reunions setmanals amb Dan Norton i les reunions de seguiment amb el tutor, s'anava avaluant el procés del projecte i es feien certes modificacions en els objectius si era adient[15].

## 3.2 SCRUM

Durant la implementació del projecte s'han agafat algunes idees bàsiques de la metodologia SCRUM per tal de dur-les a terme.

Dins la planificació s'han definit diferents sprints per realitzar les tasques. La majoria dels sprints durant una setmana, i en cada sprint es duia a terme una tasca. Cada tasca proporcionava un resultat complert, que complia amb un dels objectius fixats, per tant, es produïa un increment del producte final d'aquesta manera es podia anar mostrant com avançava el projecte. Com ja s'ha dit, setmanalment es realitzaven reunions, en aquestes reunions es mostrava el resultat obtingut en el darrer sprint, d'aquesta manera es podia donar el vist i plau a la feina feta o s'introduïen petites modificacions. Gràcies a aquests curts sprints, un canvi, no suposava la gran pèrdua de temps que podria provocar si els resultats no s'ensenyen al client fins a etapes avançades del projecte [9].

Així doncs, no s'ha aplicat la metodologia SCRUM al peu de la lletra, ja que no es disposava de tots els rols ni recursos necessaris, però si s'ha agafat aquesta idea de sprint i iterativitat de la metodologia.

## 3.3 Planificació

Per dur a terme els objectius, es defineix un conjunt de sprints curts i específics per tal de poder anar seguint la planificació amb més facilitat i complir amb els objectius. Primer de tot es va utilitzar el primer sprint per estudiar la llibreria Paper.js, i el segon per dur a terme el disseny de l'aplicació. A continuació, es poden diferenciar dues fases de implementació:

- Implementació 1: en aquesta fase s'agrupen els diferents sprints relacionats amb les tasques que ens permetran disposar d'un canvas on preparar el nostre escenari (es podrà crear una única escena).
- Implementació 2: en aquesta fase s'agrupen els diferents sprints relacionats amb les tasques que ens permetran crear diferents escenes, establir la connexió entre aquestes escenes, poder veure el resultat de l'animació, afegir el so i finalment poder descarregar l'animació.

La fase de test s'ha realitzat paral·lelament durant tota la implementació, s'han realitzat proves d'unitat de les diferents funcionalitats que es desenvolupaven.

### 3.3.1 Sprints definits

Llistat amb tots els sprints del projecte:

- Anàlisi i definició del projecte
- Estudi llibreria Paper.js
- Implementació 1.1: el software ens permet crear escenes amb diferents objectes predefinits.
- Implementació 1.2: capacitat de modificar aquests objectes en color i mida. Així com també rotar-los, moure'ls per l'escena o eliminar-los.

- Implementació 1.3: capacitat de canviar el color de fons de l'escena, funcionalitat per eliminar tota l'escena, funcionalitat per realitzar un undo (eliminar els últims canvis fets). Afegir un element nou: el llapis.
- Implementació 2.1: Poder crear noves escenes a partir de l'anterior
- Implementació 2.2: Establir connexió entre les diferents escenes, implementar que es pugui passar d'una escena a una altra a partir de l'animació necessària (moviment d'objectes, canvi de mida...). Definir el temps que transcorre entre escena i escena.
- Implementació 2.3 : Play. Un cop definides les escenes i la seva interacció s'ha de poder veure l'animació resultant.
- Implementació 2.4: Afegir àudio a les escenes. Afegir la capacitat d'afegir àudio a l'animació.
- Implementació 2.5 : Descarregar l'animació. Un cop l'usuari hagi finalitzat l'animació ha de poder descarregar-la. Aquesta descàrrega li proporcionarà el codi en llenguatge JavaScript, aquest codi l'ha de poder utilitzar en la seva pàgina web.

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Análisis i definició del projecte	21 dias	lun 8/2/16	dom 6/3/16	
2		Estudi llibreria Paper.js	3 dias	lun 7/3/16	mié 9/3/16	1
3		Disseny del software	3 dias	jue 10/3/16	lun 14/3/16	2
4		Implementació software 1.1	7 dias	mar 15/3/16	mié 23/3/16	3
5		Implementació software 1.2	7 dias	jue 24/3/16	vie 1/4/16	4
6		Implementació software 1.3	7 dias	lun 4/4/16	mar 12/4/16	5
7		Lliurament informe de progrés I	6 dias	lun 11/4/16	dom 17/4/16	
8		Implementació software 2.1	7 dias	mié 13/4/16	jue 21/4/16	6
9		Implementació software 2.2	7 dias	vie 22/4/16	lun 2/5/16	8
10		Implementació software 2.3	7 dias	mar 3/5/16	mié 11/5/16	9
11		Implementació software 2.4	7 dias	jue 12/5/16	vie 20/5/16	10
12		Lliurament informe de progrés II	6 dias	lun 16/5/16	dom 22/5/16	
13		Implementació software 2.5	7 dias	lun 23/5/16	mar 31/5/16	11
14		Test Unitari del software	59 dias	jue 10/3/16	mar 31/5/16	2
15		Realització Article del treball	8 dias	mié 1/6/16	vie 10/6/16	13
16		Test Final	8 dias	mié 1/6/16	vie 10/6/16	13

Fig. 1: Definició dels sprints amb la seva duració, començament, final i tasques predecessores[18][19]

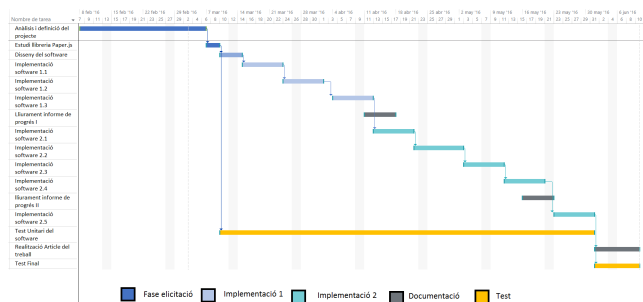


Fig. 2: Diagrama de gantt[18][19]

### 3.4 Eines utilitzades

En aquesta subsecció s'expliquen les diferents eines utilitzades durant el desenvolupament del projecte.

- JavaScript: Llenguatge de programació interpretat orientat a objectes. S'utilitza principalment en la programació web per crear pàgines web dinàmiques[7][11].
- HTML 5: 5a versió del llenguatge HTML. HTML és un llenguatge per l'elaboració de pàgines web. Defineix una estructura bàsica i un codi per la definició del contingut d'una pàgina web[7].
- CSS: Llenguatge utilitzat per definir i crear la presentació d'un document escrit en HTML[7].
- JSON: JavaScript Object Notation, és un format de text lleuger per a l'intercanvi de dades. JSON és un subconjunt de la notació literal d'objectes de JavaScript[12].
- QUnit: Framework per realitzar proves d'unitat en JavaScript. En programació, una prova d'unitat serveix per provar el correcte funcionament de cada mòdul. Serveix per assegurar que cada part del nostre software funciona correctament per separat[2][3].
- Microsoft Project: Software d'administració de projectes dissenyat per Microsoft. S'utilitza per administrar un projecte en el desenvolupament de plans, assignació de tasques, donar seguiment al progrés, administrar pressupost i analitzar càrregues de treball[20].
- Just in mind prototyper: software per realitzar prototips d'aplicacions web o mòbil[6].
- Apache: Servidor web HTTP de codi obert per plataformes Unix, Windows, Macintosh entre d'altres. Implementa el protocol HTTP/1.1 i la noció del lloc virtual.
- Atom: editor de text i codi Font lliure i de codi obert per plataformes Unix, Windows i Macintosh. És una aplicació d'escriptori construïda utilitzant tecnologies web. Té suport per una gran varietat de plugins escrits en Node.js. En aquest projecte s'han utilitzat plugins molt interessants per tal de facilitar la feina de codificació. Aquests plugins són Indentation and Beautification per la indexació del codi, auto-complete, minimap i per últim TODO-show, que ens mostra els comentaris amb paraules clau que hi ha en el projecte[21].

## 4 ARQUITECTURA DEL SISTEMA

En aquesta secció s'exposa l'arquitectura de l'aplicació web. On s'explica de quina manera s'han utilitzat les eines nomenades en la secció anterior i de quina manera estan comunicades. També s'exposa la feina desenvolupada i com està organitzada dins del projecte.

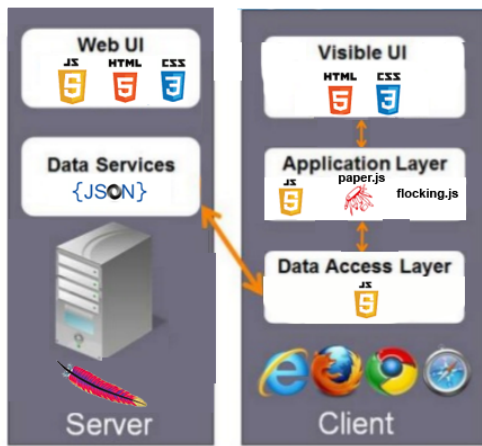


Fig. 3: Diagrama de l'arquitectura del sistema

Aquest projecte consisteix en una aplicació web. La maquetatció de la interfície gràfica d'aquesta aplicació web ha estat implementada en html5 i css3 [13]. Les funcionalitats estan implementades en el llenguatge JavaScript. Han estat utilitzades dues llibreries basades en aquest llenguatge Paper.js i Flocking.js. La primera ens permet poder crear les figures d'una manera ràpida i senzilla i tractar les seves propietats. La segona, ens permet la creació d'àudio i tractament d'aquest a partir de llenguatge JavaScript, sense necessitat de tenir pistes d'àudio en el servidor. A més a més, també ens permet modificar termes com la durada o la freqüència d'aquest àudio, podent adaptar l'àudio a la duració de l'escena corresponent. Les funcionalitats corresponents a crear una nova escena, canvi entre escenes, play de l'animació i la descàrrega de l'animació han estat desenvolupades íntegrament en aquest projecte[11]. Per realitzar la comunicació entre el servidor i el client per tal de descarregar la animació s'ha utilitzat JSON[12]. Les propietats de les escenes estan guardades en un array de JavaScript, aquest array conté totes les escenes amb totes les figures i les seves propietats en cada escena, aquest array és convertit en JSON i afegit a un fitxer. A l'hora de descarregar l'animació, es descàrrega aquest fitxer i un fitxer html amb la maquetatció necessària i al lògica necessària per importar i executar l'animació del fitxer mencionat anteriorment[10].

#### 4.1 Estructura del codi

En aquest projecte s'han desenvolupat diferents funcionalitats, aquestes funcionalitats han estat distribuïdes en diferents fitxers segons la seva lògica[17]. Aquests fitxers són els següents:

- **Index.html:** Aquest document defineix la maquetatció de l'aplicació web. L'aplicació consta únicament d'una pàgina per tant tota la codificació d'aquesta està en aquest fitxer. Hi ha dos elements canvas definits, un ens mostra les figures que podem seleccionar per la nostra animació i l'altre ens mostra l'animació creada.
- **Style.css:** Arxiu css amb les propietats de les diferents classes definides en el Index.html.
- **Figures.js:** Arxiu js que utilitza la llibreria paper.js. Aquest arxiu conté la funcionalitat per mostrar les figures que podem escollir. Implementa la crida a les funcions de creació del fitxer Animacio.js.

- **Animacio.js:** Arxiu js que utilitza la llibreria paper.js. En aquest arxiu estan desenvolupades les funcions necessàries per la creació i la modificació de les figures del canvas Animació. També està implementada la funció play, que permet anar visualitzant com va quedant l'animació mentre s'està desenvolupant.
- **Main.js:** Arxiu js amb les funcions dedicades a gestionar la creació de noves escenes, el canvi entre elles i la descàrrega final de l'animació.
- **Formularis.js:** Arxiu js amb les funcions corresponents a la validació dels formularis, les funcions són cridades des del fitxer html i passen la informació corresponent al fitxer Animacio per poder realitzar les modificacions a les figures.
- **Audios.js:** Arxiu js que utilitza la llibreria flocking.js. En aquest fitxer estan definits els diferents àudios que pot escollir l'usuari. També està definida la funció que permet engegar i parar aquests àudios.
- **kolorpicker.js:** Arxiu js on estan implementades les funcions necessàries per la paleta de colors.

## 5 DISSENY

### 5.1 Disseny funcional

En la fase de disseny és molt important definir el fluxe d'interacció de l'aplicació[14].

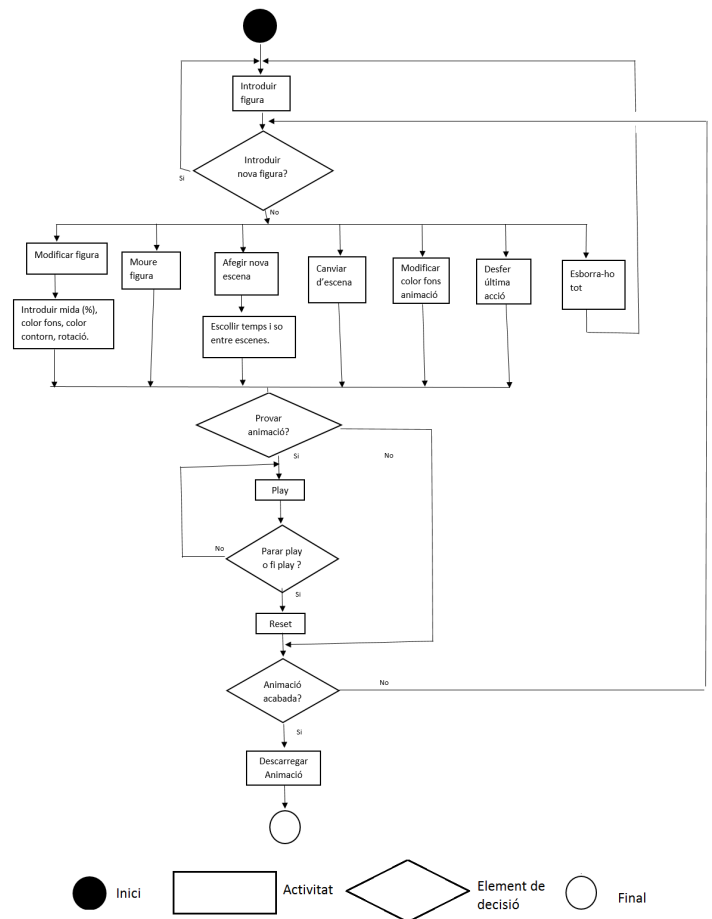


Fig. 4: Diagrama funcional

Un cop es defineix el fluxe d'interacció es té una idea de quin serà el time line de l'aplicació i com l'usuari hi interaccionarà i quines són les possibilitats que ha d'oferir segons els requeriments establerts[8].

En la figura 4 es mostra el diagrama funcional creat en la fase de disseny del software. Es mostra el fluxe d'interacció de l'aplicació. En una primera instància s'ha d'introduir una figura dintre de les possibilitats. Un cop s'ha introduït una figura podem afegir una nova figura o realitzar varies accions:

- Modificar la figura: Per dur a terme aquesta acció es mostrarà un formulari per poder escollir la mida en %, el color de fons, el color de contorn i la rotació.
- Moure la figura: canvia la posició de la figura.
- Afegir nova escena: s'haurà d'introduir el temps que es desitja que hi hagi entre l'escena anterior i la nova i també s'haurà d'escollir l'àudio que es desitgi.
- Canviar d'escena: en el canvas se'ns mostrarà l'escena que es desitgi, guardant-se els canvis de l'escena actual.
- Modificar color de fons: canviar el color de fons del canvas.
- Desfer última acció: es desfarà l'última modificació a la figura.
- Esborra-ho tot: totes les figures del canvas desapareixeran.

En el moment sigui convenient es podrà provar el resultat de l'animació amb el botó Play. Aquest botó ens mostrarà com va variant l'animació amb les diferents escenes. Un cop acabi o un cop es desitgi finalitzar amb el play, es realitza un reset. En cas d'estar finalitzada l'animació es pot descarregar, sinó, es pot continuar amb les accions ja explicades.

Un cop es garanteix que el diagrama funcional suporta tots els requisits del sistema es passa a la segona fase de disseny.

## 5.2 Disseny de la interfície

A l'hora de dissenyar com seria la interfície del software a implementar, es van tenir en compte els diferents requeriments que havia de complir el software. Un cop es van tenir clars aquests requeriments, es va realitzar un prototip inicial amb el programa Just in mind Prototyper[14][6]. En aquest prototip es va plasmar la idea d'interfície que tindria el software. Aquest prototip ha de ser coherent amb el diagrama de funcionalitat. Cada part de la interfície que permeti interaccionar amb l'usuari ha de fer valdre una acció del diagrama de funcionalitat. En aquest projecte no hi havia requisits sobre els detalls visuals de la interfície, per tant es va crear un prototip amb una interfície simple, que fos fàcil d'implementar i que fos intuïtiva per l'usuari. El prototip generat va ser el següent:

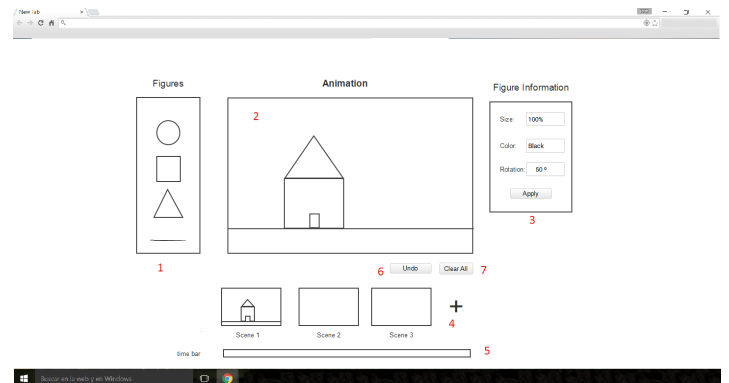


Fig. 5: 1: Conjunt de figures predefinides que l'usuari podrà escollir per fer l'animació. Fent doble clic a una de les figures es crearà una figura d'aquell tipus amb les característiques predeterminades en el canvas de l'animació. 2:Animació. Canvas on es dibuixarà l'animació, arrossegant les figures de l'animació es podrà modificar la seva posició. Fent doble clic a la figura apareixerà la informació d'aquesta. 3:Informació de la figura: Es seleccionarà una figura i sortirà la seva informació, podrà ser modificada mitjançant el formulari. 4:Conjunt d'escenes creades. Amb el botó + per afegir-ne de noves. 5:Time Bar: Barra del temps que es podrà relacionar amb les escenes, per determinar quina durada ha de tenir la animació. L'usuari podrà moure per la barra per veure com avança l'animació al llarg del temps 6:Botó de undo: Servirà per desfer l'última acció realitzada en l'animació. 7:Clear All: Esborrarà tota l'animació.

## 6 RESULTATS

En aquesta secció es mostren els resultats finals del projecte. S'expliquen els canvis realitzats respecte el disseny inicial així com les decisions d'implementació que han portat a aquesta solució.

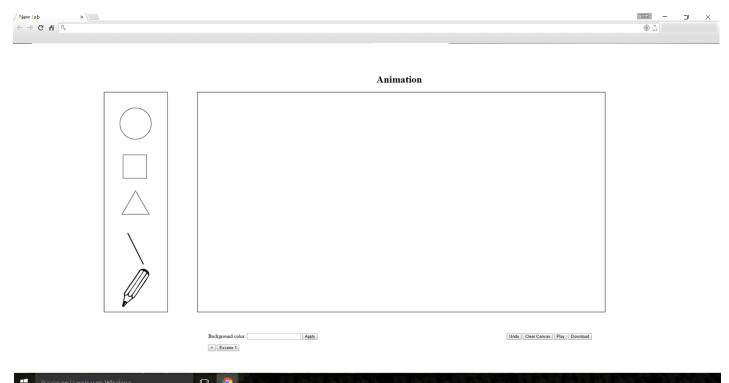


Fig. 6: Aplicació en l'estat inicial

En aquesta figura es pot contemplar la pàgina inicial, sense cap modificació. Es pot veure a l'esquerra la paleta de figures disponibles. Finalment les figures disponibles per dur a terme l'animació són figures geomètriques molt bàsiques. Una rodona, un quadrat, un triangle i una línia recta. S'ha decidit utilitzar aquestes figures ja que són les formes geomètriques més bàsiques a partir de les quals es poden crear figures més complexes. A la paleta també es pot observar un llapis, aquest llapis actua de dibuix lliure, és a dir, l'usuari podrà dibuixar el que desitgi amb el ratolí. Aquest element

però no està tractat com una figura més, ja que no serà possible modificar-lo durant les diferents escenes. Està pensat per dibuixar un possible fons estàtic de l'animació, per donar més joc i poder crear animacions amb més diversitat. Es pot escollir el color del llapis. Un canvi important respecte el disseny inicial és l'eliminació de la barra de temps. En l'aplicació final l'usuari pot tenir diverses escenes i aquestes escenes guarden l'estat de les figures en aquell instant, l'usuari podrà anar movent-se entre les diferents escenes per veure els canvis que es realitzen. Per tal que l'usuari pugui veure l'animació de canvi, s'ha introduït el botó Play. Al pitxar aquest botó l'animació transcorrerà des de l'escena 1 fins l'última i es podran veure tots els canvis en les figures. Aquests canvis poden ser:


- Canvi de mida: la figura creixerà o disminuirà de mida al llarg de l'animació.
- Canvi de posició: la figura es mourà del punt inicial fins el final passant pel punt que es troba en totes les escenes.
- Canvi de color: el canvi de color es realitzarà a partir d'una interpolació entre els dos colors. Aquesta interpolació està implementada a través dels valors RGB de les figures.
- Canvi de orientació: la figura rotarà de l'orientació inicial fins la final passant per totes les orientacions de les escenes.

Tot hi haver modificat el disseny actual es segueixen complint tots els requisits del sistema, l'usuari pot anar veient com evoluciona la seva animació. Un cop ha vist com queda i vol seguir-hi treballant només caldrà donar-li al botó de reset (el botó de play haurà passat a ser de reset).

Un altre requisit era el fet de poder modificar una figura, aquesta modificació es realitza a partir d'un formulari que apareix quan es fa doble clic sobre la figura.

### Rectangle information:

Figure size (%):

Fill color:  


Line color:  

Figure rotation(degrees):

Fig. 7: Formulari de modificació de la figura

En aquesta figura es mostra el formulari per modificar una figura. La mida de la imatge es modifica introduint el % que es vol modificar. Per canviar el color de fons o el de contorn, s'introdueix el codi hexadecimal o es pot escollir en una paleta de colors (figura 8). L'orientació es calcula en graus. En el formulari hi ha dos botons, el primer, modify, aplica les modificacions introduïdes, el segon, remove figure, elimina la figura.

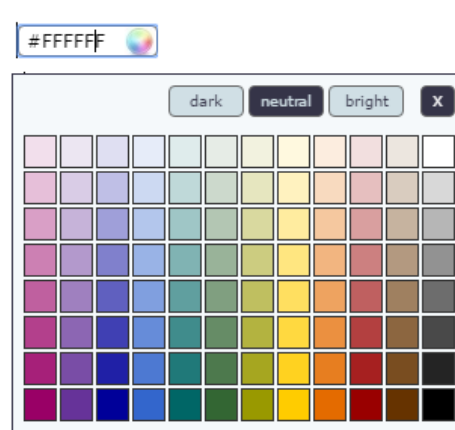


Fig. 8: Paleta de colors

Quan es vol crear una nova escena, és necessari introduir el temps de transició entre l'escena anterior i la nova i l'àudio que hi haurà entre aquestes dues escenes. Aquestes dades s'introdueixen a partir d'un popup que apareix quan pitxem el botó +.

**Enter new scene**

Enter scene time (10-250):

Enter sound scene

Fig. 9: Popup escena nova

En aquesta figura es mostra el popup esmentat. El temps de transició ha d'estar comprès entre 10 i 250 unitats. L'àudio es pot escollir entre un llistat ja definit. El botó play serveix per escoltar l'àudio escollit en aquell moment. El botó accept crea finalment la nova escena amb el temps i l'àudio introduïts.

En la figura 6 es pot visualitzar a sota del canvas de l'animació diversos elements. El primer, Background color, serveix per determinar el color de fons de l'animació. També podem veure els botons undo, clear canvas i play, ja comentats anteriorment. Per últim el botó Download, al clicar en aquest botó es descarrega l'animació realitzada. Es descarreguen dos arxius:

- Animation.js: Aquest arxiu conté la informació de totes les figures en totes les animacions. La informació



està guardada en JSON de manera que pot ser importada desde un fitxer html.

- **Animation.html:** Aquest arxiu conté la descripció de un canvas on es veu l'animació creada. Per tal de poder veure l'animació creada té un petit script que llegeix el JSON del fitxer Animation.js i executa la funció play per veure l'animació.

Així doncs, un usuari que vol utilitzar l'animació creada en el seu lloc web només caldrà que introdueixi el fitxer Animations.js en el directori, disposi d'un element canvas en el seu fitxer html i enganxi el script en el seu fitxer html.

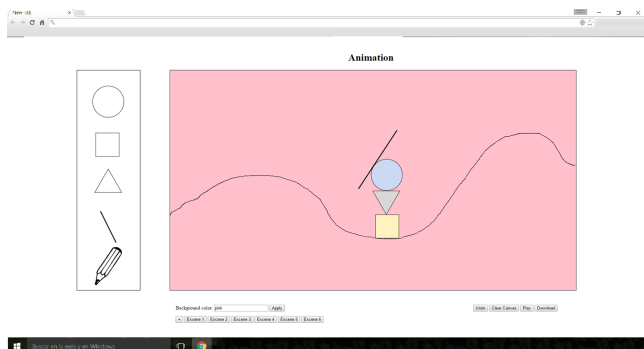


Fig. 10: Estat de l'aplicació amb una animació creada

En aquesta figura s'observa una animació ja finalitzada, que té 6 escenes, s'ha modificat el color de fons, hi ha una figura de cada tipus i s'ha utilitzat l'eina del llapis.

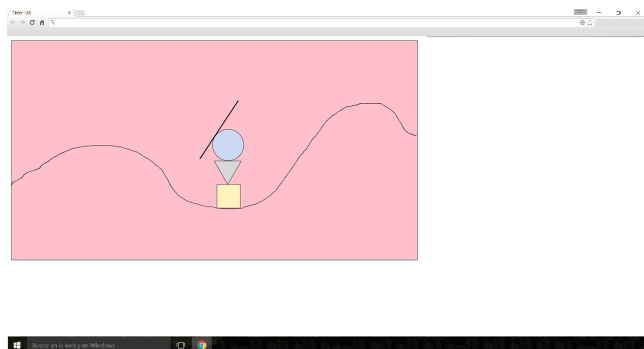


Fig. 11: Fitxer Animation.html obtingut de la descarrega de l'animació

Per últim, aquest és l'aspecte que té l'arxiu Animation.html per l'animació anterior. Simplement constà d'un canvas amb l'animació creada, per tant, aquest canvas pot ser introduït a qualsevol lloc web i es disposaria de l'animació creada en un altre lloc web.

## 7 POSSIBLES MILLORES

Aquest projecte a complert amb els objectius inicials, tot i així, hi ha certs aspectes que podrien millorar el projecte i fer-lo més funcional i més atractiu pels usuaris.

- Una primera millora podria ser a nivell visual, degut al poc temps per realitzar el projecte s'ha prioritzat el aspecte funcional per sobre del visual, i no s'han gastat

grans recursos en com es veu el software. Com ja s'ha comentat, no hi havia requisits de detalls visuals, per tal de millorar aquest fet es podria fer una nova fase de disseny amb noves idees que fessin el software més atractiu.

- Per altre banda, un fet que afegiria molt joc al software seria la possibilitat d'afegir al programari la gestió d'usuaris. Dins la mateixa aplicació web una persona podria crear-se un usuari i accedir-hi. Aquest usuari podria guardar les seves animacions i al tornar a iniciar sessió més endavant podria recuperar-les i modificar-les. Aquesta millora suposaria la creació i gestió d'una base de dades simple. El software ja existent no hauria de ser modificat, simplement s'haurien d'afegir les noves funcionalitats. A la base de dades es guardarien les dades bàsiques de l'usuari i per aquest es guardarien les diferents animacions que té, de cada animació simplement es guardaria el fitxer JSON generat. Quan l'usuari volgués carregar l'animació es podria realitzar un import d'aquest fitxer i l'usuari podria seguir amb la seva edició.
- Afegir noves figures dintre de la paleta. Una opció que també donaria més joc a l'aplicació és la possibilitat que l'usuari escollir entre més varietat de figures.
- Possibilitat d'importar arxius d'àudio propis. En la mateixa línia també es podria incorporar una petita graella per l'edició i creació de pistes d'àudio. D'aquesta manera l'usuari podria crear les seves pròpies o modificar les ja existents.
- Per últim, es podria transformar en una aplicació multi-plataforma, de manera que tingués la seva versió també per dispositius smartphone o tabletas. Aquesta transformació es podria dur a terme amb phonegap. Phonegap és un framework per desenvolupar aplicacions mòbils a partir d'eines web com html5, JavaScript o css. D'aquesta manera podríem utilitzar el codi ja existent i amb una única versió tenir una aplicació multiplataforma.

## 8 CONCLUSIONS

- El projecte ha avançat d'acord amb els objectius marcats a l'inici i el producte final és satisfactori. S'ha aconseguit crear una aplicació web on es poden crear animacions amb elements digitals que interactuen a partir dels canvis que ha introduït l'usuari. Aquesta animació es pot descarregar i el codi obtingut és JavaScript, per tant, l'animació es pot utilitzar en altres llocs web.
- Tot i haver patit alguns canvis d'objectius a meitat del projecte amb el seu corresponent impacte en la planificació s'ha pogut adaptar i reorganitzar el treball gràcies a anar treballant en petits sprints i gràcies a tenir un feedback constant. Ha estat molt important realitzar reunions setmanals per obtenir aquest feedback i gràcies a utilitzar una metodologia iterativa basada en sprints els canvis que s'han hagut de dur a terme han pogut ser realitzats sense un impacte major que suposés el fracàs del projecte.

- Més enllà de complir amb les necessitats per les quals es va crear el projecte, el projecte és interessant i pot ser utilitzat per totes aquelles persones que vulguin crear petites animacions a partir de figures digitals sense necessitat de tenir coneixements de programació. Això és degut a que s'ha creat un producte que no existia al mercat i compleix una necessitat amb la que els usuaris d'Internet es poden trobar avui en dia.
- El fet que JavaScript dia rere dia vagi guanyant terreny a Flash pel que fa a l'animació web dóna un argument més per creure que aquest projecte és viable i pot arribar a convertir-se en una eina de moda.
- A nivell personal el projecte a resultat molt enriquidor i ha suposat un repte. He aprofundit en els meus coneixements sobre la programació web, més específicament en el camp d'animacions web mitjançant JavaScript, un camp molt complex i interessant i que està en constant creixement.

## AGRAÏMENTS

Agrair l'ajuda rebuda per part del tutor Fernando Vilariño durant tot el projecte, proporcionant-me informació que em podia ser útil i donant idees sobre la implementació i el disseny d'aplicació. També agrair la immensa ajuda i enriquiment que ha proporcionat al projecte l'artista Dan Norton. Donant feedback de la feina feta i aportant idees noves.

## REFERÈNCIES

- [1] Zack Grossbart. How to create web animations with Paper.js. Smashingmagazine.
- [2] Victor Tejada Yau. QUnit para nuestro código JavaScript. Desarrolladorweb.com
- [3] Pàgina oficial del framework QUnit: <http://www.qunitjs.com>
- [4] <http://www.paperjs.org/>
- [5] Pàgina web de desenvolupament de Dan Norton: <http://www.ablab.org/>
- [6] <http://www.justinmind.com/overview>
- [7] Pàgina web d'informació per desenvolupadors web. Conté tutorials i referències a temes de desenvolupament web com HTML, CSS, JavaScript: <http://www.w3schools.com/>
- [8] Ingeniería de software, Ian Sommerville, Pearson (Mèxic) (26 d'abril de 2012).
- [9] Métodos Ágiles y Scrum, Alonso Álvarez García, Rafael de las Heras del Dedo, Carmen Lasagómez, Manuales imprescindibles (23 de gener de 2012).
- [10] Desarrollo de aplicaciones web, Carlos Mateu, Catalunya: Fundación para la Universitat Oberta de Catalunya (Març de 2004).
- [11] JavaScript, La guía definitiva, David Flanagan, Anaya Multimedia (2007).
- [12] Introducción a JSON: <http://www.json.org/json-es.html>
- [13] Diseño y desarrollo web con html5 y css, VV.AA (2011).
- [14] Gestión del proceso software, VV.AA(2002).
- [15] Software: Requisitos de calidad y ergonomía, VV.AA(2005).
- [16] Pàgina oficial de la llibreria flocking.js: <http://flockingjs.org/>
- [17] Clean code: a handbook of agile software craftsmanship. R.C. Martin. Prentice Hall, 2008 and Anaya Multimedia, 2012.
- [18] Rapid Development: Taming Wild Software Schedules, Steve McConnell, 1996.
- [19] Software Estimation: Demystifying the Black Art (Developer Best Practices), Steve McConnell, Paperback ( Març del 2006).
- [20] Software de administración de proyectos, Microsoft Project: <https://products.office.com/es-es/project/project-and-portfolio-management-software>
- [21] 10 Essential Atom Add-ons: <https://www.sitepoint.com/10-essential-atom-add-ons/>